Network Working Group	G. Zorn, Ed.
Internet-Draft	Network Zen
Intended status: Informational	Q. Wu
Expires: January 12, 2012	T. Taylor
	Huawei
	K. Hoeper
	Motorola
	S. Decugis
	Free Diameter
	Y. Nir
	Check Point
	July 11, 2011

Handover Keying (HOKEY) Architecture Design
draft-ietf-hokey-arch-design-04

Abstract

The Handover Keying (HOKEY) Working Group seeks to minimize handover delay due to authentication when a peer moves from one point of attachment to another. Work has progressed on two different approaches to reduce handover delay: early authentication (so that authentication does not need to be performed during handover), and reuse of cryptographic material generated during an initial authentication to save time during re-authentication. A starting assumption is that the mobile host or "peer" is initially authenticated using the Extensible Authentication Protocol (EAP), executed between the peer and an EAP server as defined in RFC 3748.
This document documents the HOKEY architecture. Specifically, it describes design objectives, the functional environment within which handover keying operates, the functions to be performed by the HOKEY architecture itself, and the assignment of those functions to architectural components. It goes on to illustrate the operation of the architecture within various deployment scenarios that are described more fully in other documents produced by the HOKEY Working Group.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The Extensible Authentication Protocol (EAP) [RFC3748] is an authentication framework that supports different types of authentication methods. Originally designed for dial-up connections, EAP is now commonly used for authentication in wireless access networks.

When a host (or "peer", the term used from this point onward) changes its point of attachment to the network, it must be re-authenticated. If a full EAP authentication must be repeated, several message round-trips between the peer and the home EAP server may be involved. The resulting delay will result in degradation or in the worst case loss of any service session in progress if communication is suspended while re-authentication is carried out. The delay is worse if the new point of attachment is in a visited network rather than the peer's home network, because of the extra procedural steps involved as well as because of the probable increase in round-trip time.

[RFC5169] describes this problem more fully and establishes design goals for solutions to reduce re-authentication delay for transfers

within a single administrative domain. [RFC5169] also suggests a number of ways to achieve a solution:

- specification of a method-independent, efficient, re-authentication protocol;
- reuse of keying material from the initial authentication;
- deployment of re-authentication servers local to the peer to reduce round-trip delay; and
- specification of the additional protocol needed to allow the EAP server to pass authentication information to the local re-authentication servers.

[RFC5295] tackles the problem of reuse of keying material by specifying how to derive a hierarchy of cryptographically independent purpose-specific keys from the results of the original EAP authentication. [RFC5296] specifies a method-independent re-authentication protocol (ERP) applicable to two specific deployment scenarios:

- where the peer's home EAP server also performs re-authentication; and
- where a local re-authentication server exists but is collocated with a AAA proxy within the domain.

Other work provides further pieces of the solution or insight into the problem. For the purpose of this draft, [RFC5749] provides an abstract mechanism for distribution of keying material from the EAP server to re-authentication servers. [RFC5836] contrasts the EAP re-authentication (ER) strategy provided by [RFC5296] with an alternative strategy called "early authentication". [RFC5836] defines EAP early authentication as the use of EAP by a mobile peer to establish authenticated keying material on a target attachment point prior to its arrival. Here, a full EAP execution occurs before the handover of the peer takes place. Hence, the goal of EAP early authentication is to complete all EAP-related communications, including AAA signaling, in preparation for the handover, before the mobile device actually moves. Early authentication includes direct and indirect pre-authentication as well as Authenticated Anticipatory Keying (AKK). All three mechanims provide means to execute a full EAP authentication with a Candidate Access Point (CAP) while still being connected to the Serving Access Point (SAP) but vary in their respective system assumptions and communication paths. In particular, direct pre-authentication assumes that clients are capable of discovering candidate access points and all communications are routed through the serving access point. On the other hand, indirect pre-authentication assumes an existing relationship betweem SAP and CAP, whereas in AAK the client interacts with the AAA to discover and connect to CAPs.

Both EAP re-authentication and early authentication enable faster inter-authenticator handovers. However, it is currently unclear how the necessary handover infrastructure is deployed and can be integrated into existing EAP infrastructures. In particular, previous work has not described how ER servers that act as endpoints in the re-authentication process should be integrated into local and home domain networks. Furthermore, it is currently unspecified how EAP infrastructure can support the timely triggering of early authentications and aid with the selection of candidate access points.

This document proposes a general HOKEY architecture and demonstrates how it can be adapted to different deployment scenarios. To begin with, Section 3 recalls the design objectives for the HOKEY architecture. Section 4 reviews the functions that must be supported within the architecture. Section 5 describes the components of the HOKEY architecture. Finally, Section 6 describes the different deployment scenarios that the HOKEY Working Group has addressed and the information flows that must occur within those scenarios, by reference to the documents summarized above where possible and otherwise within this document itself.

2. Terminology

This document contains no normative language, hence [RFC2119] language does not apply.

This document reuses most of the terms defined in Section 2.2 of [RFC5836]. In addition, it defines the following:

EAP Early Authentication
: The use of EAP by a mobile peer to establish authenticated keying material on a target attachment point prior to its arrival, see [RFC5836].

EAP Re-authentication (ER)
: The use of keying material derived from an initial EAP authentication to enable single-roundtrip re-authentication of a mobile peer. For a detailed description of the keying material see Section 3 of [RFC5296].

ER Server
: A component of the HOKEY architecture that terminates the EAP re-authentication exchange with the peer.

ER Key Management
: An instantiation of the mechanism provided by [RFC5749] for creating and delivering root keys from an EAP server to an ER server.

3. Design Goals

This section investigates the design goals for the HOKEY architecture. These include reducing the signaling overhead for re- authentication and early authentication, integrating local domain name discovery, and improving deployment scalability. These goals supplement the discussion in [RFC5169].

3.1. Reducing Signalling Overhead

3.1.1. Minimized Communications with Home Servers

ERP requires only one round trip, however, this roundtrip may require communications between a peer and its home ER and/or home AAA server in explicit bootstrapping and communication between local servers and home server in Implicit bootstrapping even if the peer is currently attached to a visited (local) network. As a result, even this one round trip may introduce long delays because home ER and home AAA servers may be distant from the peer and the local server to which the peer is attached. To lower the signaling overhead, communication with the home ER server and home AAA server should be minimized. Ideally, a peer should only need to communicate with local servers and other local entities.

3.1.2. Minimized User Interaction for authorization

When the peer is firstly attached to the network or moves between heterogeneous networks, normally EAP full authentication between the peer and EAP server occurs and User Interaction for authorization may be needed, e.g., a dialog is prompted to the user for a personal identifier. To lower the signaling overhead, user interaction for authorization at each time of handover should be minimized. Ideally, user interaction once for authorization and then transparently authenticating in other places during handover is a desirable solution which also can be used to improve user experience.

3.1.3. Integrated Local Domain Name (LDN) Discovery

ERP bootstrapping must occur before (implicit) or during (explicit) a handover to transport the necessary re-authentication root keys to the local ER server involved. Implicit bootstrapping is preferable because it does not require communication with the home ER server during handover (see section 3.1.1), but it requires the peer to know the domain name of the ER server before subsequent local ERP exchange happens in order to derive the necessary re-authentication keying material. [RFC5296] does not specify such a domain name discovery mechanism and suggests that the peer may learn the domain name through the EAP- Initiate/ Re-auth-Start message or via lower layer announcements. However domain name discovery happens after the implicit bootstrapping completes, which potentially may introduce extra latency.

To allow more efficient handovers, a HOKEY architecture should support an efficient domain name discovery mechanism and allow its integration with ERP implicit bootstrapping. Even in the case of explicit bootstrapping, local domain name discovery should be optimized such that it does not require contacting the home AAA server, as is currently the case.

3.2. Better Deployment Scalability

To provide better deployment scalability, it should not be required that the HOKEY server and AAA servers or proxies are collocated. Separation of these entities may cause problems with routing, but allows flexibility in deployment and implementation.

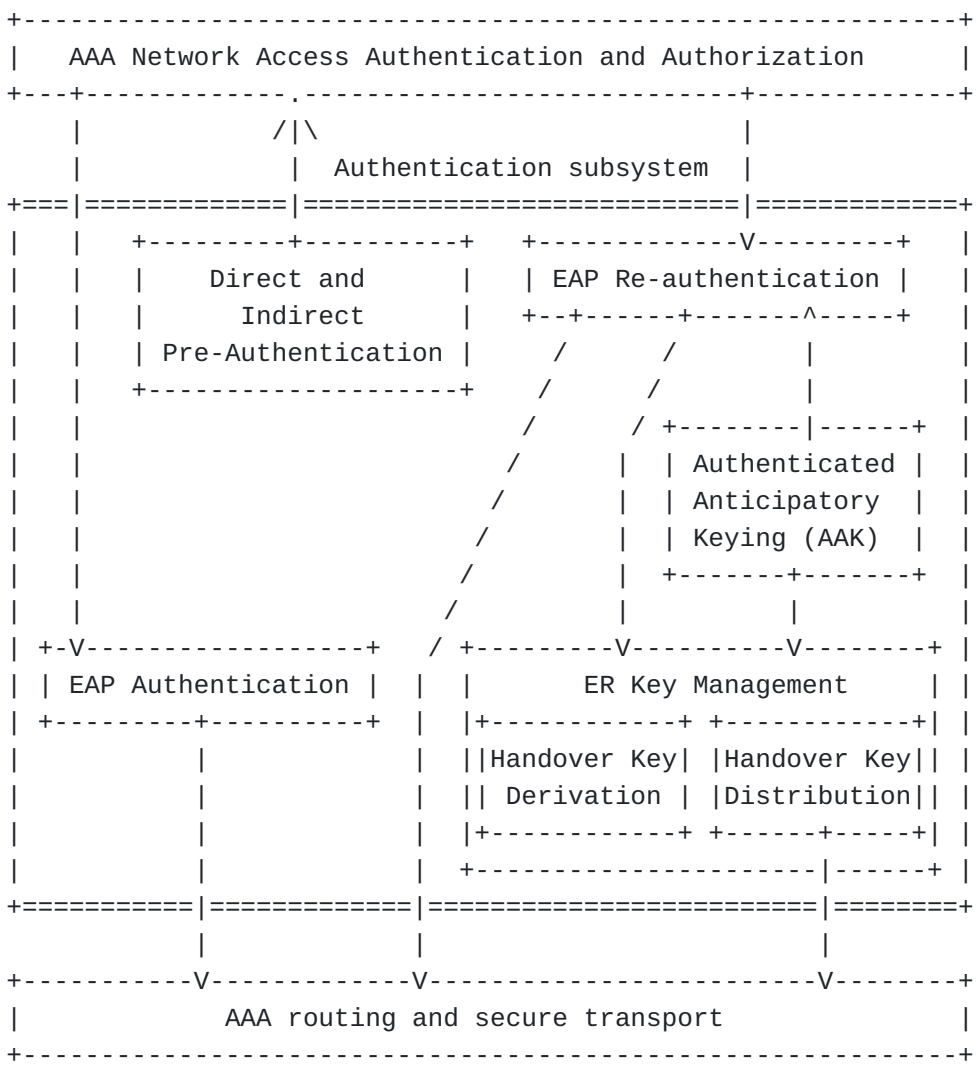
4. Functions That Must Be Supported

4.1. System Overview

This section views the HOKEY architecture as the implementation of a subsystem providing authentication services to AAA. Not only does AAA depend on the authentication subsystem, but the latter also depends on AAA as a means for the routing and secure transport of messages internal to the operation of network access authentication.
The operation of the authentication subsystem also depends on the availability of a number of discovery functions:

- discovery of candidate access points, by the peer, by the serving attachment point, or by some other entity;
- discovery of the authentication services supported at a given candidate access point;
- discovery of the required server in the home domain when a candidate access point is not in the same domain as the serving attachment point, or no local server is available;
- peer discovery of the local domain name (LDN) when EAP re-authentication is used with a local server.

It is assumed that these functions are provided by the environment within which the authentication subsystem operates, and are outside the scope of the authentication subsystem itself. Local domain name discovery is a possible exception.
Figure 1 shows the major functions comprising the authentication subsystem and their interdependencies. These functions are described below. [EDITOR'S NOTE: These probably need refinement. The relationship of pre-authentication to EAP authentication, for instance, is currently not totally correct, when one takes account of the roles described in Section 5. AAK also needs an extension of ER key management.]



Arrows show the direction of functional dependency.
Figure 1 shows the following dependencies:

*When AAA is invoked to authenticate and authorize network access, it uses one of two services offered by the authentication subsystem: full EAP authentication, or EAP re-authentication.

*Pre-authentication triggers AAA network access authentication and authorization at each candidate access point, which in turn causes full EAP authentication to be invoked.

*EAP re-authentication invokes ER key management at the time of authentication to create and distribute keying material to ER servers.

*Authenticated anticipatory keying (AAK) relies on ER key management to establish keying material on ER/AAK servers, but uses an extension to ER key management to derive and establish

keying material on candidate authenticators. Also AAK uses an extension to EAP re-authentication to communicate with ER/AAK servers.

EAP authentication, EAP re-authentication, and handover key distribution depend on the routing and secure transport service provided by AAA. Discovery functions and the function of authentication and authorization of network entities (access points, ER servers) are not shown. As stated above, these are external to the authentication subsystem.

4.2. Pre-Authentication Function (Direct or Indirect)

The pre-authentication function is responsible for discovery of candidate access points and completion of network access authentication and authorization at each candidate access point in advance of handover. The operation of this function is described in general terms in [RFC5836]. No document is yet available to describe the implementation of pre-authentication in terms of specific protocols. [RFC5873] could be part of the solution, but is Experimental rather than Standards Track.

4.3. EAP Re-authentication Function

The EAP re-authentication function is responsible for authenticating the peer at a specific access point using keying material derived from a prior full EAP authentication. [RFC5169] provides the design objectives for an implementation of this function. [RFC5296] describes a protocol to implement EAP re-authentication subject to the architectural restrictions noted above. Work is in progress to relax those restrictions.

4.4. EAP Authentication Function

The EAP authentication function is responsible for authenticating the peer at a specific access point using a full EAP exchange. [RFC3748] defines the associated protocol. [RFC5836] shows the use of EAP as part of pre-authentication. Note that the HOKEY Working Group has not specified the non-AAA protocol required to transport EAP frames over IP that is shown in Figures 3 and 5 of [RFC5836], although [RFC5873] is a candidate.

4.5. Authenticated Anticipatory Keying (AAK) Function

The authenticated anticipatory keying function is responsible for pre-placing keying material derived from an initial full EAP authentication on candidate access points. The operation is carried out in two steps: ER key management (with trigger not currently specified) places root keys derived from initial EAP authentication onto an ER/AAK server associated with the peer. When requested by the peer, the ER/AAK server

derives and pushes predefined master session keys to a list of candidate access points. The operation of the authenticated anticipatory keying function is described in very general terms in [RFC5836]. A protocol implementation is being specified in [I-D.ietf-hokey-erp-aak].

4.6. EAP-Based Handover Key Management

EAP-based handover key management consists of EAP method independent key derivation and distribution and comprises the following specific functions: [RFC5295], and key distribution is specified in [RFC5749].

- *handover key derivation; and
- *handover key distribution.

The derivation of handover keys is specified in

5. Components of the HOKEY Architecture

This section describes the components of the HOKEY architecture, in terms of the functions they perform. The components cooperate as described in this section to carry out the functions described in the previous section. Section 6 describes the different deployment scenarios that are possible using these functions.
The components of the HOKEY architecture are as follows:

- *the peer;
- *the authenticator, which is a part of the serving access point and candidate access points;
- *the EAP server; and
- *the ER server, and
- *the ER/AAK server , [I-D.ietf-hokey-erp-aak]either in the home domain or local to the authenticator.

5.1. Functions of the Peer

The peer participates in the functions described in Section 4 as shown in Table 1.

Function	Peer Role
EAP authentication	Determines that full EAP authentication is needed based on context (e.g., initial authentication), prompting from the authenticator, or discovery that

Function	Peer Role
	only EAP authentication is supported. Participates in the EAP exchange with the EAP server.
-	-
Direct pre-authentication	Discovers candidate access points. Initiates pre-authentication with each, followed by EAP authentication as above, but using IP rather than L2 transport for the EAP frames.
-	-
Indirect pre-authentication	Enters into a full EAP exchange when triggered, using either L2 or L3 transport for the frames.
-	-
EAP re-authentication	Determines that EAP re-authentication is possible based on discovery or authenticator prompting. Discovers ER server. Participates in ERP exchange with ER server.
-	-
Authenticated anticipatory keying	Determines that AAK is possible based on discovery or serving authenticator prompting. Discovers candidate access points. Sends request to serving authenticator to distribute keying material to the candidate access points.
-	-
ER key management	No role.

Functions of the Peer

5.2. Functions of the Serving Authenticator

The serving authenticator participates in the functions described in Section 4 as shown in Table 2.

Function	Serving Authenticator Role
EAP authentication	No role.
-	-
Direct pre-authentication	No role.
-	-
Indirect pre-authentication	Discovers candidate access points. Initiates an EAP exchange between the peer and the EAP server through each candidate authenticator. Mediates between L2 transport of EAP frames on the peer side and a non-

Function	Serving Authenticator Role
	AAA protocol over IP toward the candidate access point.
-	-
EAP re-authentication	No role.
-	-
Authenticated anticipatory keying	Mediates between L2 transport of AAK frames on the peer side and AAA transport toward the ER/AAK server.
-	-
ER key management	No role.

Functions of the Serving Authenticator

5.3. Functions of the Candidate Authenticator

The candidate authenticator participates in the functions described in Section 4 as shown in Table 3.

Function	Candidate Authenticator Role
EAP authentication	Invokes AAA network access authentication and authorization upon handover/initial attachment. Mediates between L2 transport of EAP frames on the peer link and AAA transport toward the EAP server.
-	-
Direct pre-authentication	Invokes AAA network access authentication and authorization when the peer initiates authentication. Mediates between non-AAA L3 transport of EAP frames on the peer side and AAA transport toward the EAP server.
-	-
Indirect pre-authentication	Same as direct pre-authentication, except that it communicates with the serving authenticator rather than the peer.
-	-
EAP re-authentication	Invokes AAA network access authentication and authorization upon handover. Discovers or is configured with the address of the ER server. Mediates between L2 transport of a ERP frames on the peer side and AAA transport toward the ER server.
-	-
	Receives and saves pMSK.

Function	Candidate Authenticator Role
Authenticated anticipatory keying	
-	-
ER key management	No role.

Functions of the Candidate Authenticator

5.4. Functions of the EAP Server

The EAP server participates in the functions described in Section 4 as shown in Table 4.

Function	EAP Server Role
EAP authentication	Authenticates and authorizes the candidate access point to act as authenticator. Terminates EAP signalling between it and the peer via the candidate authenticator. Determines whether network access authentication succeeds or fails. Provides MSK to authenticator.
-	-
Direct pre-authentication	As for EAP authentication.
-	-
Indirect pre-authentication	As for EAP authentication.
-	-
EAP re-authentication	Mutually authenticates with the ER server and authorizes it for receiving keying amterial. Provides rRK or DSrRK to the ER server.
-	-
Authenticated anticipatory keying	As for EAP re-authentication.
-	-
ER key management	Creates rRK or DSrRK and distributes it to ER server requesting the information.

Functions of the EAP Server

5.5. Functions of the ER Server

The ER server participates in the functions described in Section 4 as shown in Table 5. [EDITOR'S NOTE: Need discussion of respective roles

of local and home ER server, or whether there should even be such a distinction.]

Function	ER Server Role
EAP authentication	No role.
-	-
Direct pre-authentication	No role.
-	-
Indirect pre-authentication	No role.
-	-
EAP re-authentication	Authenticates and authorizes the candidate access point to act as authenticator. Authenticates itself to the EAP server and acquires rRK or DSrRK as applicable when necessary. Terminates ERP signalling between it and the peer via the candidate authenticator. Determines whether network access authentication succeeds or fails. Provides MSK to authenticator.
-	-
Authenticated anticipatory keying	Authenticates itself to the EAP server and acquires rRK or DSrRK as applicable when necessary. Authenticates and authorizes the candidate access points to act as authenticator. Derives pMSKs and passes them to the candidate access points.
-	-
ER key management	Receives and saves rRK or DSrRK as applicable.

Functions of the ER Server

6. Usage Scenarios

Depends on whether it involves a change in a domain or access technology, we have the following the usage scenarios.

6.1. Intra-domain handover

The peer moves between two authenticators in the same domain. In this scenario, the peer communicates with the ER server via the ER authenticator within the same network.

6.2. Inter-domain handover

The peer moves between two different domains. In this scenario, the peer communicates with more than one ER servers via one or two different ER authenticators. One ER server is located in the current network as the peer, one is located in the previous network from which the peer moves. Another ER server is located in the home network which the peer belong to.

6.3. Inter-technology handover

The peer moves between two heterogeneous networks. In this scenario, The peer needs to support at least two access technologies. The coverage of two access technologies usually is overlapped during handover. In this case, only authentication corresponding to intra-domain handover is required, i.e., the peer can communicates with the same local ER server to complete authentication and obtain keying materials corresponding to the peer.

7. AAA Consideration

This section provides an analysis of how the AAA protocol can be applied for hokey architecture in accordance with Authentication Subsystem Functional Overview in figure 1.

7.1. Authorization

Authorization is a major issue in deployments. Wherever the peer moves around, the home AAA server provides authorization for the peer during its handover. However authorization is not necessary to couple with authentication at each time of handover, since authorization is only needed when the peer is firstly attached to the network or moves between two different AAA domains. The EAP Key management document [RFC5247] discusses several vulnerabilities that are common to handover mechanisms. One important issue arises from the way the authorization decisions which might be handled at the AAA server during network access authentication. For example, if AAA proxies are involved, they may also influence in the authorization decision. Furthermore, the reasons for choosing a particular decision are not communicated to the AAA clients. In fact, the AAA client only knows the final authorization result. Another issue is about session management. In some circumstance when the peer moves from one authenticator to another, the peer may be authenticated by the different authenticator during a period of time and the authenticator to which the peer is currently attached needs to create new AAA user session, however the AAA Server should not view these handoffs as different sessions. Otherwise this may affect user experience and also accounting or logging issues. For example, the session-Id creation, in most case, is done by each authenticator to which the peer attaches. In this sense, the new authenticator acting as

AAA Client needs to create new AAA user session from scratch which forces its corresponding AAA Server to terminate the existing user session with previous authenticator and setup the new user session with the new authenticator. This may complicate the set up and maintenance of the AAA User session.

7.2. Transport aspect

The existing AAA protocols can be used to carry EAP messages and ERP messages between the AAA server and AAA clients AAA Transport of ERP messages is specified in [RFC5749] and [I-D.ietf-dime-erp]. AAA transport of EAP message is specified in [RFC4072].The key transport also can be performed through a AAA protocol. [I-D.ietf-dime-local-keytran] specifies a set of Attribute-Value Pairs (AVPs) providing native Diameter support of cryptographic key delivery.

8. IANA Considerations

This document does not require any actions by IANA.

9. Acknowledgments

The authors would like to thank Mark Jones, Zhen Cao,Lionel Morand for their reviews of previous versions of this draft.

10. References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC3748]	Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J. and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
[RFC5169]	Clancy, T., Nakhjiri, M., Narayanan, V. and L. Dondeti, "Handover Key Management and Re-Authentication Problem Statement", RFC 5169, March 2008.
[RFC5295]	Salowey, J., Dondeti, L., Narayanan, V. and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)", RFC 5295, August 2008.
[RFC5296]	Narayanan, V. and L. Dondeti, "EAP Extensions for EAP Re-authentication Protocol (ERP)", RFC 5296, August 2008.
[RFC5749]	Hoeper, K., Nakhjiri, M. and Y. Ohba, "Distribution of EAP-Based Keys for Handover and Re-Authentication", RFC 5749, March 2010.
[RFC5836]	Ohba, Y., Wu, Q. and G. Zorn, "Extensible Authentication Protocol (EAP) Early Authentication Problem Statement", RFC 5836, April 2010.

[RFC5873]	Ohba, Y. and A. Yegin, "Pre-Authentication Support for the Protocol for Carrying Authentication for Network Access (PANA)", RFC 5873, May 2010.
[I-D.ietf-hokey-erp-aak]	Cao, Z, Deng, H, Wang, Y, Wu, Q and G Zorn, "EAP Re-authentication Protocol Extensions for Authenticated Anticipatory Keying (ERP/AAK)", Internet-Draft draft-ietf-hokey-erp-aak-06, October 2011.
[RFC5247]	Aboba, B., Simon, D. and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework", RFC 5247, August 2008.
[I-D.ietf-dime-erp]	Bournelle, J, Morand, L, Decugis, S, Wu, W and G Zorn, "Diameter support for EAP Re-authentication Protocol (ERP)", May 2011.
[I-D.ietf-dime-local-keytran]	HuaweiNetwork Zen, "Diameter Attribute-Value Pairs for Cryptographic Key Transport", July 2010.
[RFC4072]	Eronen, P., Hiller, T. and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", RFC 4072, August 2005.

Authors' Addresses

Glen Zorn editor Zorn Network Zen 227/358 Thanon Sanphawut Bang Na, Bangkok 10260 Thailand Phone: +66 (0) 87-040617 EMail: gwz@net-zen.net

Qin Wu Wu Huawei Technologies Co.,Ltd Site B, Floor 12F, Huihong Mansion, No.91 Baixia Rd. Nanjing, JiangSu 210001 China Phone: +86-25-84565892 EMail: sunseawq@huawei.com

Tom Taylor Taylor Huawei Technologies Co., Ltd Ottawa, Canada EMail: tom111.taylor@bell.net

Katrin Hoeper Hoeper Motorola, Inc. 1301 E. Algonquin Road Schaumburg, IL 60196 USA EMail: khoeper@motorola.com

Sebastien Decugis Decugis Free Diameter 4-2-1 Nukui-Kitamachi Tokyo, Koganei 184-8795 Japan EMail: sdecugis@freediameter.net

Yoav Nir Nir Check Point 5 Hasolelim st. Tel Aviv 67897 Israel EMail: ynir@checkpoint.com